# 数据级权限控制

这里的实现方式分为对对数据表和视图数据集的获取和最终页面资源分配两个方面

## 对数据表和视图的数据集的获取进行控制,是通过在定义自定义表管理时进行设置的

自定义表管理的定义过程示意图如下所示:



### 选择数据源

数据源指的是”系统数据源”。

### 选择表或视图

在“选择数据源”中选到的数据源中选择一表或视图。

### 设置显示字段

指定要在最终页面进行显示的表/视图的列。

### 设置过滤条件

过滤是针对选择的数据表/视图的数据集进行过滤。过滤条件的定义方式有图形接口和脚本(Groovy 脚本)两种。

### 定义自定义变量



定义任意的变量，并可在模板或条件脚本中使用。

自定义变量的值可以是固定值、来自脚本、来自页面的输入。

自定义的变量最终被放入一个HashMap 类型的容器VarMap中。

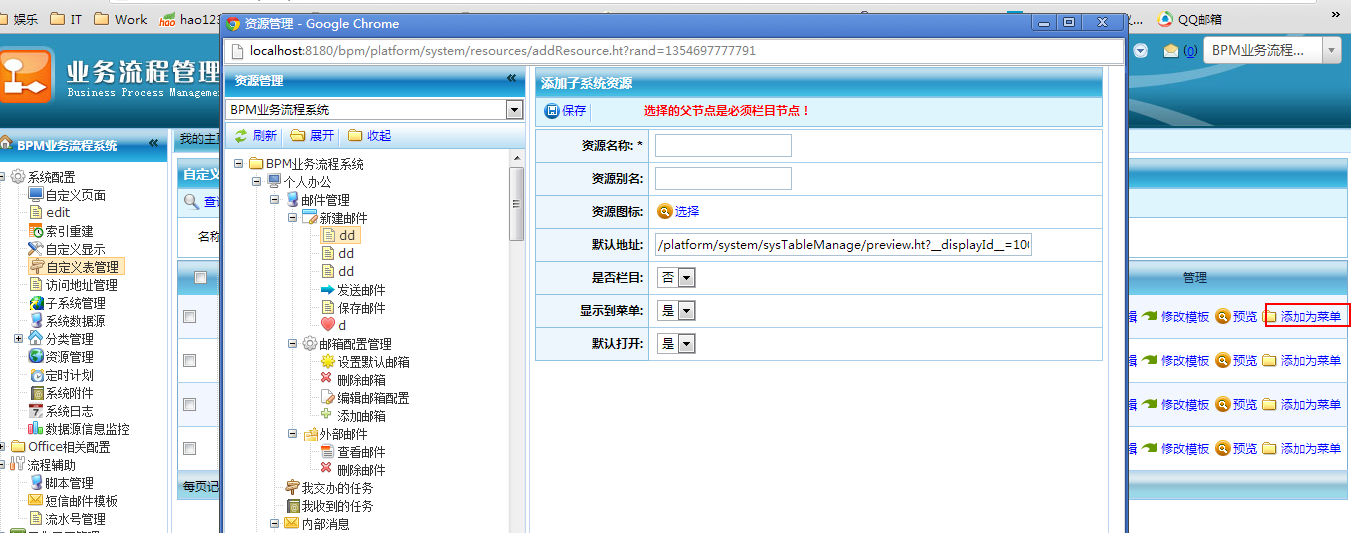
### 选择数据模板

选择的数据集和设置最终要通与数据模板结合才能形成最终的显示页面。这里的数据模板可以在自“自定义表单”模板中进行定义。

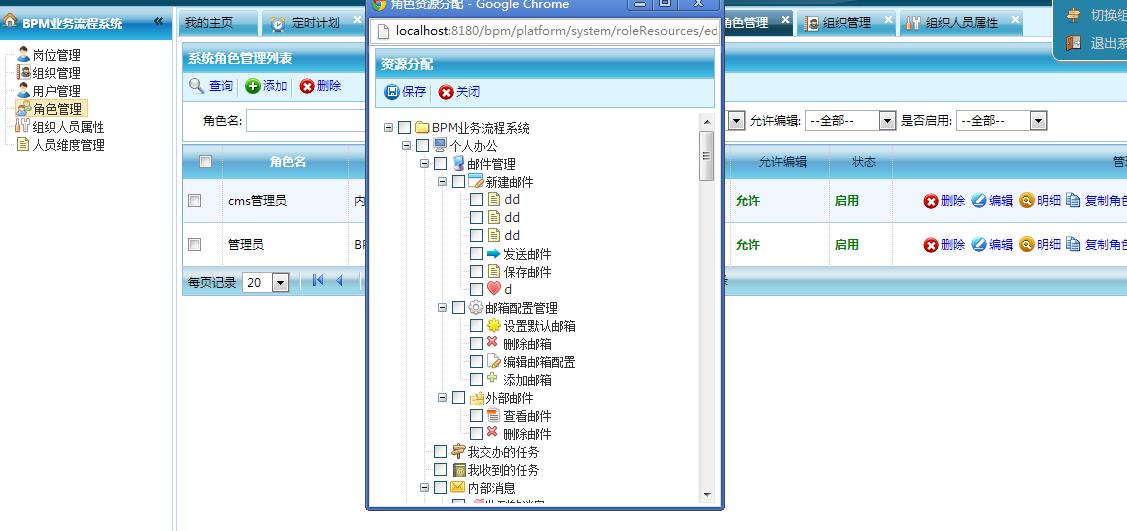
## 最终页面资源分配.

定义的自定义表管理可作为一资源进行分配:

如下:



将自定表管理页面定义为菜单资源后,即可在角色管理中,进行资源分配.



自定义表管理实现过程是 以表/视图的元数据+FreeMarker模板通过两次解析最终生成一个Html的页面。

# 使用方法：

## 是否分页：使用的是服务器端分页。

分页：能过分页方式从表/视图从数据库中取得数据集并以分页方式显示。

不分页：取数据、显示时都不分页。



## 是否可编辑 （只对有主键的数据表有效）

可编辑：最终生成的页面只是显示

不可编辑：最终生成的页面带有“管理”一列，可对数据进行编辑和删除。





## 显示字段设置

在“显示”栏中勾选要显示的数据列。在“注释”栏中，设置数据列的注释，注释将作为字段的最终的字段显示名。



### 过滤条件设置：

#### 图形接口方式设置

设置的条件字段对应到取得在数据库中取数据的SQL WHERE条件

概念说明：

联合类型：包括AND、OR

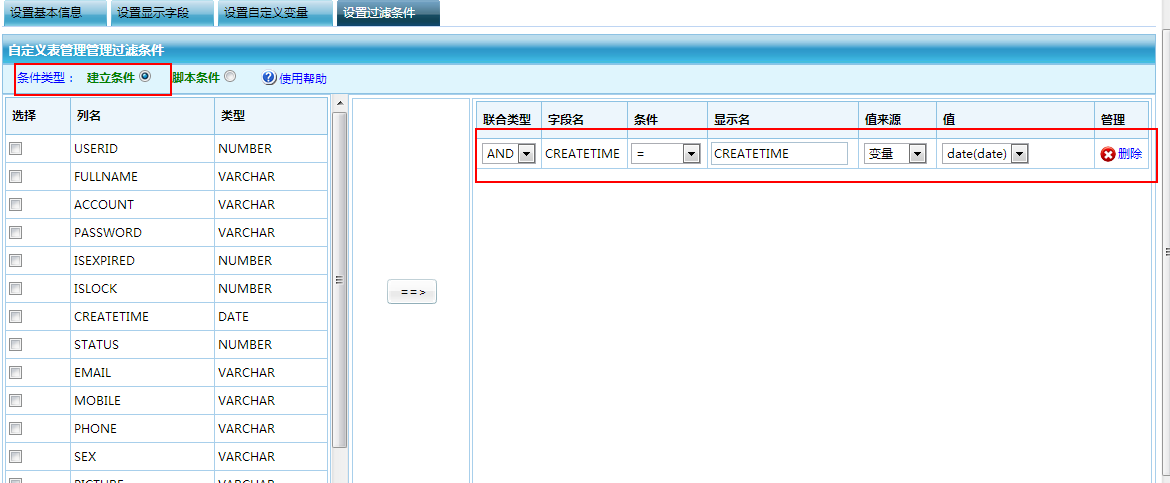
条件：条件的比较类型

显示名：当值来源为“输入”时，在页面中的显示名称。

值来源：固定值- 直接输入固定的值

脚本-值来自脚本

变量-值来自自定义变量



#### 脚本方式设置

当要定义一些复杂且使用图形接口方式无法完成时，可以使用使用脚本方式定义过滤条件。脚本指的Groovy脚本 。

编写的脚本要求，脚本在最终解析后，得到一个如**"col1=value1 and (col2=value2 or col2=value3)"**,可用于**SQL**语句的**WHERE**子句的字符串。 在脚本中，可以通过**VarMap**访问自定义的变量。**VarMap**是一个**HashMap**类型的变量，对**VarMap**操作与**Java**中对**HashMap**类型的变量相同。 例如，要添加一个动态的用户ID过滤条件，如果自定义变量中的定义了变量userId，且值不为空，则将userId作为过滤条件:  
String wh=**""**;  
Long userId=VarMap.get("userId");  
**if**(userId!=**null**){  
    wh="userId="+userId;  
}  
**return** wh;



## 自定义变量定义

### 概念

自定义变量是用户自行添加的变量。变量与变量的值作为一个Key、Value对被放入到一个HashMap的容器中，容器的名字约定为VarMap。并通过VarMap进行访问，访问方式与在Java中访问HashMap类型的变量相同。

### 变量定义

自定义变量包括定义变量的名字、注释、类型、值来源、和值。名字命名规则与Java语言中的变量相同。

自定义变量的值来源可分为：

输入，此类型的变量要求在FreeMarker模板在定义一个对应的输入框供用户输入。变量的值最终从页面的提交请求中获取。

固定值，此类型的变量的值在定义变量时直接指定。

脚本，脚本类型的值在变量使用时动态解析脚本时得到。

### 使用变量

自定义的变量可以在FreeMarker模板和过滤条件中使用。使用时通过预定义HashMap类型的容器变量VarMap进行访问。

如：

在FreeMarker中访问自定义变量name：<#assisn na=VarMap["name"] >

在脚本类型的过滤条件中访问自定义变量name：Object na=VarMap.get("name");

